CMSI 371-01
COMPUTER GRAPHICS
Spring 2016

Assignment 0329a Feedback

This is the last assignment where *3a* tops out at **|** as we head toward the full expected functionality of your library code. Meanwhile, *3d* also stays at **|** until we get sufficient functionality in our shaders.

Chris Franco *cf7 / spe861@gmail.com*

*Notes while running (high-priority notes are marked with ***):*

- (looking at the latest version of the scene) Some interesting notions here…but pertaining to the scope of this assignment, a variety of shapes is seen. They may or may not be implementing the child/group functionality, so we'll need to look at the code for that.
- Shape unit test suite seen, and it runs successfully.

Code review (refer to http://lmucs.github.io/hacking-guidelines/ for code-review abbreviations):

1. Good collection of polygon meshes implemented here, although your cone appears to be missing a triangle, and it's hollow. (*+1b*, *+3a*)
2. *** The basic child/group data structure is there, but as it turns out it is not handled correctly by the drawing code. The sequence in lines 237–257 of *hello-webgl-again.js* is a bit of a cheat: instead of respecting the true tree nature of the child/group functionality, the tree is "flattened" into the top-level `objectsToDraw` array. The remainder of the code then treats the scene as a straight-up list of objects. This results in duplication of labor down the line: e.g., if your scene interaction involves the dynamic removal/addition of objects to the scene, if that object is the child of another, then the code needs to know to also remove/add this object from/to the `objectsToDraw` array. Better to manipulate the object tree directly, then have everything "just work." In a sense, this implementation only goes halfway. (*1c*, *4a*, *4b*)

1b — **+**

1c — **/** …Partway there, but the top-level processing/drawing sequence ignores the composition tree.

3a (max **|**) — **|**

3d (max **|**) — **|**

4a — **|** …Functional for now, with some gaps lurking in the future.

4b — **|** …Same reason: tree-processing limitations.

4c — **+**

4d — **+**

4e — **+** …Excellent frequency, descriptive messages.

4f — **+** …Sufficient work done by the due date.